Katedra informatiky
Přírodovědecká fakulta
Univerzita Palackého v Olomouci

BAKALÁŘSKÁ PRÁCE

Generátor náhodných příběhů

2022

Jakub Mazanec

Vedoucí práce: Mgr. Tomáš Kühr, Ph.D.

Studijní obor: Aplikovaná informatika, prezenční forma

Bibliografické údaje

Autor:	Jakub Mazanec
Název práce:	Generátor náhodných příběhů
Typ práce:	bakalářská práce
Pracoviště:	Katedra informatiky, Přírodovědecká fakulta, Univerzita Palackého v Olomouci
Rok obhajoby:	2022
Studijní obor:	Aplikovaná informatika, prezenční forma
Vedoucí práce:	Mgr. Tomáš Kühr, Ph.D.
Počet stran:	31
Přílohy:	1 CD/DVD
Jazyk práce:	český

Bibliograhic info

Author:	Jakub Mazanec
Title:	Random story generator
Thesis type:	bachelor thesis
Department:	Department of Computer Science, Faculty of Science, Palacký University Olomouc
Year of defense:	2022
Study field:	Applied Computer Science, full-time form
Supervisor:	Mgr. Tomáš Kühr, Ph.D.
Page count:	31
Supplements:	1 CD/DVD
Thesis language:	Czech

Anotace

Cílem práce je navrhnout a implementovat softwarový nástroj pro generování jednoduchých příběhů. Program by měl být schopen na základě zadaných slov, slovních spojení a modelů vět sestavit i rozsáhlejší anglický text a při tomto zohlednit standardní koncepty (zápletka, rozuzlení, závěr, kapitola, odstavec, atd.) z literární teorie i běžná gramatická pravidla.

Synopsis

Goal of this work is to develop software tool for generating simple stories. Program should be capable to create a complex English text based on imputed words and sentence models. It should follow classical writing concepts (paragraphs, story structure) as well as grammatical rules of English language.

Klíčová slova: programování; spisovatelství; generátor

Keywords: programming; writing; generator

Chtěl bych poděkovat vedoucímu práce Tomášovi Kührovi za asistenci na této práci.

Místopřísežně prohlašuji, že jsem celou práci včetně příloh vypracoval/a samostatně a za použití pouze zdrojů citovaných v textu práce a uvedených v seznamu literatury.

datum odevzdání práce podpis autora

Obsah

Seznam obrázků

Seznam zdrojových kódů

1 Význam generátorů textu

1.1 Generativní umění

Generativní umění je umění, které z nějaké části vzniklo náhodně. Autor tohoto díla tedy navhl nějaký mechanizmus, který sám stvoří aspoň nějakou část onoho díla. Většina tohoto textu bude věnována generátorům digitálním, ovšem lze najít i manuálně řízené způsoby.

V roce 1792 v Berlíně vznikla generovaná hudební hra, která byla připsána autoru Wolfgangovi Amadeovi Mozartovi. Pomocí hodu kostkami byli zvoleny náhodné přednapsané fragmenty hudby. Po spojení náhodně vybraných prvků pak vzniklo hotové dílo.[1]

1.2 Nahrazení lidské práce

Už dnes jsou dostupné a některými firmamy využívané generátory na bázy Artificial Inteligence (AI) pro psaní textu. [2] Důvodem je zde velké množství velmi repetetivní práce. Například psaní popisu velkého množství produktů, které si jsou navzájem velmi podobné, často vede ke frusraci pro lidského zaměstnance. Boti se v tomto dají využít pro vytvoření různorodých popisů na základě několika klíčových slov pro jednotlivé produkty. Zdroj [2] uvádí, že zákazníci mají pozitivnější reakce na popis psaný v běžné řeči, než jen výpis klíčových slov. Další využití může být psaní reklamních sloganů, příspěvků na sociálních sítích, nebo předpověď počasí podle naměřených dat.

2 Úvod do generátorů textů

Pod název generátor textu se dá zařadit několik různých typů generátoru a každý z nich má trochu jiné využití. V této sekci si rozebereme ty základní.

2.1 Lorem ipsum

Asi nejjednodužší náhodně generovaný text je Lorem ipsum (Lipsum), který se tváří jako nějaký cizý jazyk. Většinou připomíná latinu. Lipsum se používá například při vytváření prototipů webových stránek nebo při designu fontů. Důvod pro použití Lipsum v takových případech se uvádí ten, že v případě použití smysluplného textu, člověka rozptyluje obsah textu od posuzování kvality grafického designu. Lipsum je i preferováno proti textu, který byste dostali pomocí rychlého mačkání písmen na klávesnici, protože Lipsum svím rozvržením mezer a kombinací souhlásek a samohlásek připomíná běžné jazyky.

Na internetu lze najít mnoho stránek které nabízejí různé verze generátorů Lipsum a nabízejí i různé specifikace jako třeba délka textu nebo začátek textu, protože Lipsum text často začíná frází „Lorem ipsum“, aby čtenář věděl, že se nejedná o text v cizím jazyce.[3]

2.2 Umělá inteligence

Strojové učení je užitečný nástroj pro generování náhodných textů. Inteligenci je předloženo nějaké množství předlohy. Umělá inteligence se pak následně snaží vypozorovat v předloze nějaké vzorce a je trénována, aby vytvářela texty které jsou ideálně co nejvíce se tváři jako své předlohy. (Samozdřejmě bez toho, aby přímo kopírovala předlohu.)[4]

2.2.1 Přirozené zpracování jazyka

V angličntině označované jako Natural Language Processing (NLP). Je to věda zajímající se o inteakci mezi lidskými jazyky a počítači. [5] S využitím NLP vývojáři mohou provádět funkce jako sumarizace, rozčlenění témat, překlad, oprava gramatických chyb nebo rozpoznání textu.NLP chápe text ne jako sekvenci symbolů, ale jako slova, sousloví a věty, které mají svůj význam.

NLP je komplikovaná věda. My lidé rozumíme lidské řeči od ranného dědství, ovšem lidský jazyk je často nepřesný a vyžaduje dobré chápání jak jazyka tak našeho světa ke správné interpretaci. Většinou při NLP je řeč o strojovém učení, ale spadají zde i ručně psanná pravidla o jazyce.

2.2.2 Botnik

Generátory textu založené na umělé inteligenci mají často problém s pochopením, jak funguje reálný svět a tak jejich texty jsou plné velmi atypických situací. Pro ilustraci použiji úryvek z knihy Harry Potter And A Portrait Of What Looked Like A Large Pile Of Ash od umělé inteligence Botnik. [6]

„Ron was standing there and doing a kind of frenzied tap dance. He saw Harry and immediately began to eat Hermione's family."

V českém překladu:
„Tam stál Ron a prováděl nějaký zběsilý stepovací tanec. Zpatřil Harryho a okamžitě začal jíst Hermioninu rodinu."

Ve zbytku kapitoly není nijak zmíněno, co dělali Hermioniny rodiče ve škole čar a kouzel nebo proč se Ron dal na kanibalismus. Toto nepochopení reálného světa generátory textu může být výhodou, pokud chceme, aby generátor generoval absurdní komedii, ale pokud chceme nějaký vážnější žánr, například horor, narazíme na velký problém.

2.2.3 OpenAI

Tato umělá inteligence dosahuje velmi dobrých výsledků při porozumění obsahu textu. [7] Jedná se v tomto případě o více, než jen generátor textu. Pokud například jí předložíte úryvek textu, OpenAI dokáže zodpovědět dotazy k onomu textu nebo třeba zestručnit ho. Toto pochopení významu textu vede mimo jiné ke generování textů, které jsou často nerozeznatelné od textů, psaných lidmi.

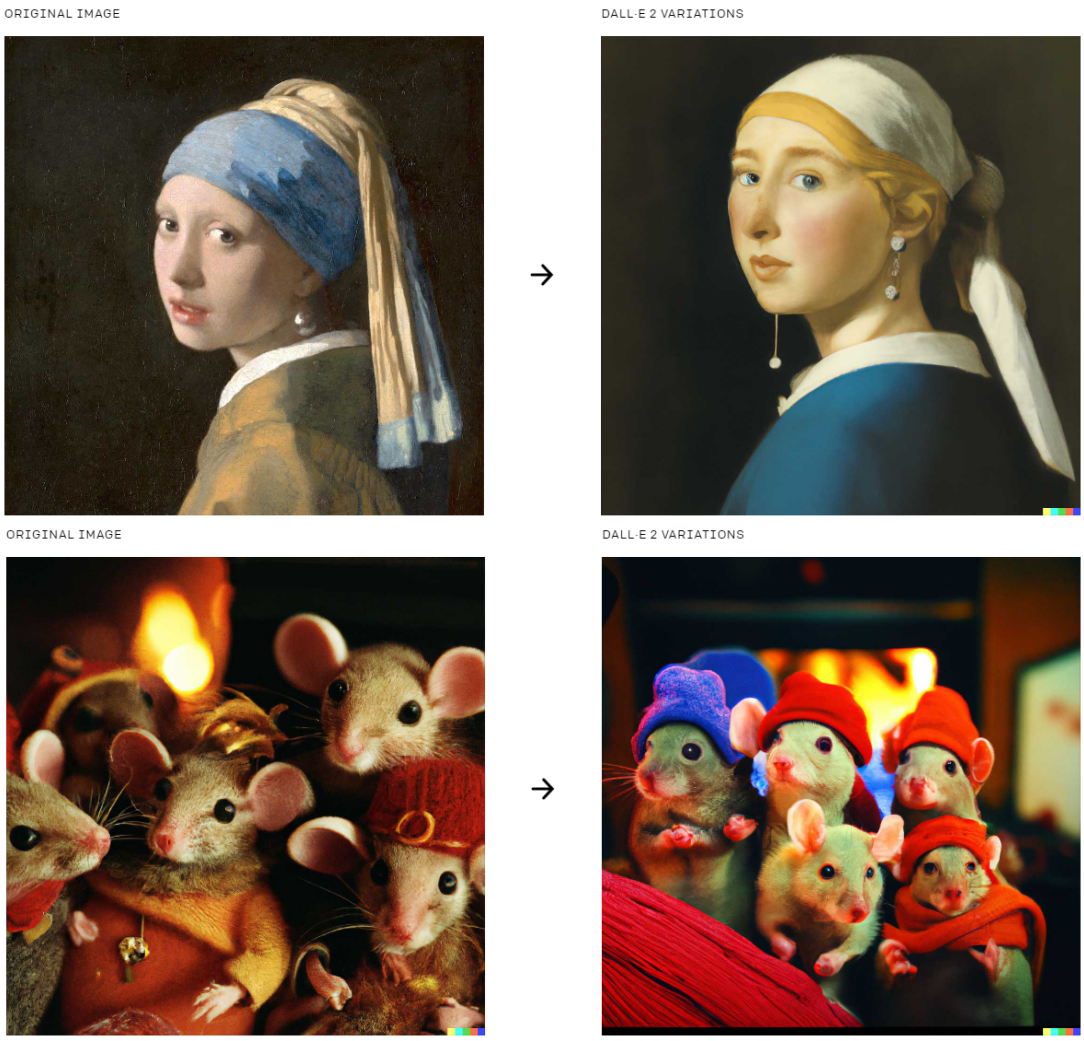
Zde je ukázka textu na požadavek „Napiš příběh odehrávající se na Universitě Palackého“.

„Je to příběh o tom, jak se naše hrdinka setkala se svým osudem. Jmenovala se Klára a byla to studentka prvního ročníku na Univerzitě Palackého. Před několika měsíci, když se Klára chystala na univerzitu, její rodiče zemřeli v tragické nehodě. Byli to její největší opory a ona se cítila strašně sama. Jakmile Klára dokončila první rok na univerzitě, byla si jista, že chce dělat něco smysluplného se svým životem. Rozhodla se, že se bude věnovat charitě a pomáhat těm, kteří to potřebují. Začala pracovat v dobročinné organizaci, která se zaměřovala na pomoc studentům na univerzitách.

Jednoho dne, když Klára pracovala v kanceláři, přišel dovnitř mladý muž. Byl to student druhého ročníku, jmenoval se Tomáš. Tomáš byl viditelně nervózní a Klára se ho snažila uklidnit. Tomáš ji řekl, že jeho rodiče také zemřeli před několika měsíci a že se cítí strašně sám. Řekl, že potřebuje pomoc, protože se chystá skončit se životem. Klára ho okamžitě chytila za ruku a řekla mu, že mu pomůže. Tomášovi se nakonec podařilo zůstat naživu a díky Kláře měl nový smysl života.“

Jak je z ukázky zřejmé, že OpenAI dokáže pracovat i v češtině a jiných jazycích. Je však dobré zmínit, že ne vždy dobře. Často na zadání napsané v češtině odpovídá anglicky, nebo třeba při psaní textu v Esperantu, jedno slovo nahradila jeho anglickým překladem. Je dále velmi pozitivní, že OpenAI při mém testování zvládá i velmi dobře gramatiku anglickou, i českou.

OpenAI disponuje i dalšími vlastnostmi. Dokáže například převádět prostou řeč do programovacího jazyka nebo zodpovídat otázky ze světa, jako například kdo byl Václav Havel a podobně. Projekt OpenAI dal za vznik i systému DALL · E 2, což je nástroj pro generování obrázků nebo jejich úpravu. [8] DALL · E 2 je scopen generovat realistické obrázky nebo kreslit obrazy inspirované vzorem. DALL · E 2 proces kreslení začíná v náhodně rozmístěným zrněním na „plátně“ a postupnými iteracemi hledá vhodné vzory a postupně zrnění přetvoří v čistý obraz.



Obrázek 1: ukázka obrázků vygenerovaných skrze DALL · E 2

Slabou stránkou OpenAI mi příjde humor. Píši „příjde“ samozdřejmě z důvodu, že kvalita humoru je subjektivní. Většina umělých inteligencí dosahují humoru vytvářením absurdních situací. Tyto absurdní situace jsou důsledek nepochopení přesného významu slov a světa jako takového. OpenAI však tato chápání má mnohem lepší, a tak má humor spíš založený na údernosti, pokud to vůbec za humorné považujeme. Zde je pár vtipů vygenerovaných na požadavek „Napiš vtip o Univerzitě Palackého.“. V této práci při vybírání ukázek většinou vybírám ty nejzajímavější vzorky, ale zde, pro demonstrování běžné kvality, vypíšu pět po sobě jdoucích výsledků.

„Univerzita Palackého je jako žena: nevíš, jestli je to dobrá nebo špatná, protože se to každý den mění.“

„Palacký University is so old, it was founded by Julius Caesar!“

„Univerzita Palackého je tak stará, že když ji založili, už tu byla Praha.“

„Na Palackého univerzitě se říká, že když se na ní někdo učí, tak se učí celý život.“

„Univerzita Palackého je ve skutečnosti kouzelnická škola.“

2.2.4 Sunspring

Sunspring je krátký film jehož scénář byl napsán umělou inteligencí jménem Jetson. Je plně dostupný na platformě YouTube[9]. Jako podklad pro učení dostal kolem 100 sci-fi scénářů.

Většina textu se zdá nesmyslná, ale divák si dokáže najít nějaký výklad toho textu, pokud ho tam chce vidět. Některé fráze třeba znějí jako nějaká nová rčení. Například taková fráze „I'm little bit of a boy on a floor.“ Význam jednotlivích vět je však na interpretaci diváka.

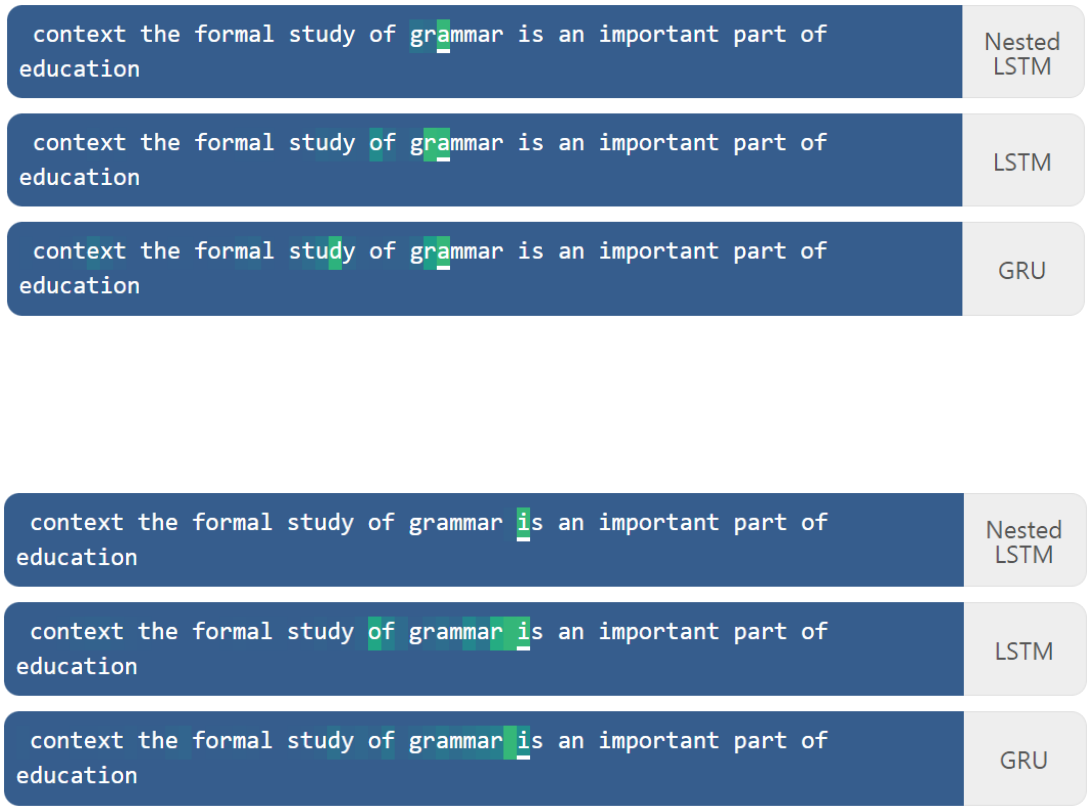
Co jsem pročítal komentáře a i moje vlastní interpretace filmu se tím shoduje, tak zápletka filmu je taková, že žena opustila jednoho muže kvůli druhému. (V tom filmovém zpracování je to tedy žena. Ve scénáři se o „ní“ referuje jako „he“, tedy česky „on“)

Ne jenom divák si musí sám interpretovat ten film, ale i samotná režie si musela interpretovat některé části. Herci sice můžou říci cokoliv, ale zahrát například scénu kde postava „...is standing in the stars and sitting on the floor.“ vyžaduje svou interpretaci.

2.2.5 Automatické doplňování textu

Mezi využití textových generátorů spadá i automatické doplňování textu. AI se v tomto případě snaží co nejlépe předpovídat další písmeno, nebo napsat následující slovo.

Při řešení této problematiky se řeší otázka krátkodobé paměti versus dlouhodobé paměťi. Článek [10] pojednává o této problematice v rámci technologie Recurent Neural Network (RNN). Jedná se o odnož strojového učení, která na základě vstupní sekvence generuje pokračování. [11] Na jednu stranu krátkodobá paměť (třeba předchzí slovo) by měla mít větší důraz, protože více pomáhá indikovat následující slovo nebo znak. Ovšem dlouhodobá paměť je také důležitá, protože dává širší kontext pro předvídání. Kdy tedy brát v potaz vzdálenější části řetězce? Obvzláště při předvídání jednotlivých znaků je dobré nehodnotit algoritmus pouze podle přesnosti předpovědí, protože algoritmus může třeba jen nabrat dobré hodnocení na dokončování slova, pokud je již z části rozepsané, ale vůbec nebere v potaz jak se slova ovlivňují v rámci věty nebo jestli věta dává logický smysl.



Obrázek 2: vyzualizace důležitosti písmen při předvídání textu

Obrázek 2 je vyzualizace ze článku [10] který znázorňuje jak rozdílné algoritmy přiřazují důraz při předpovídání jednotlivých písmen. Článek umožňuje inteaktivně zobrazit důležitost pro všechna písmena, ale já zde pro ukázku použiji jen první písmeno „a“ ve slově „grammar“ a „i“ ze slova „is“.

Long-Short-Term Memory (LSTM) a Gated Recurrent Units (GRU) jsou algoritmy pracující na podobné bázy, ale rozdílem je, že GRU má paměťové buňky zkládající se ze vstupní brány a brány zapomenutí, zatímco LSTM má navíc bránu výstupní. [12, 13] GRU má paměťové buňky, které mají také méně parametrů. Uvádí se, že GRU dosahuje podobně dobrých výsledků kvality, jako LSTM, a je tedy více efektivnější. Nested LSTM pak je nadstavbou nad LSTM, jelikož má své paměťové buňky strukturované v komplexnějších strukturách, což vede k lepší reprezentaci dat. [14]

2.3 Algoritmy bez strojového učení

Existují i mnohé algoritmy pro generování textu, které mají veškerou logiku přímo naprogramovanou programátorem, a nikoliv strojově naučenou podle předloh. Praktická část této práce také spadá do této kategorie.

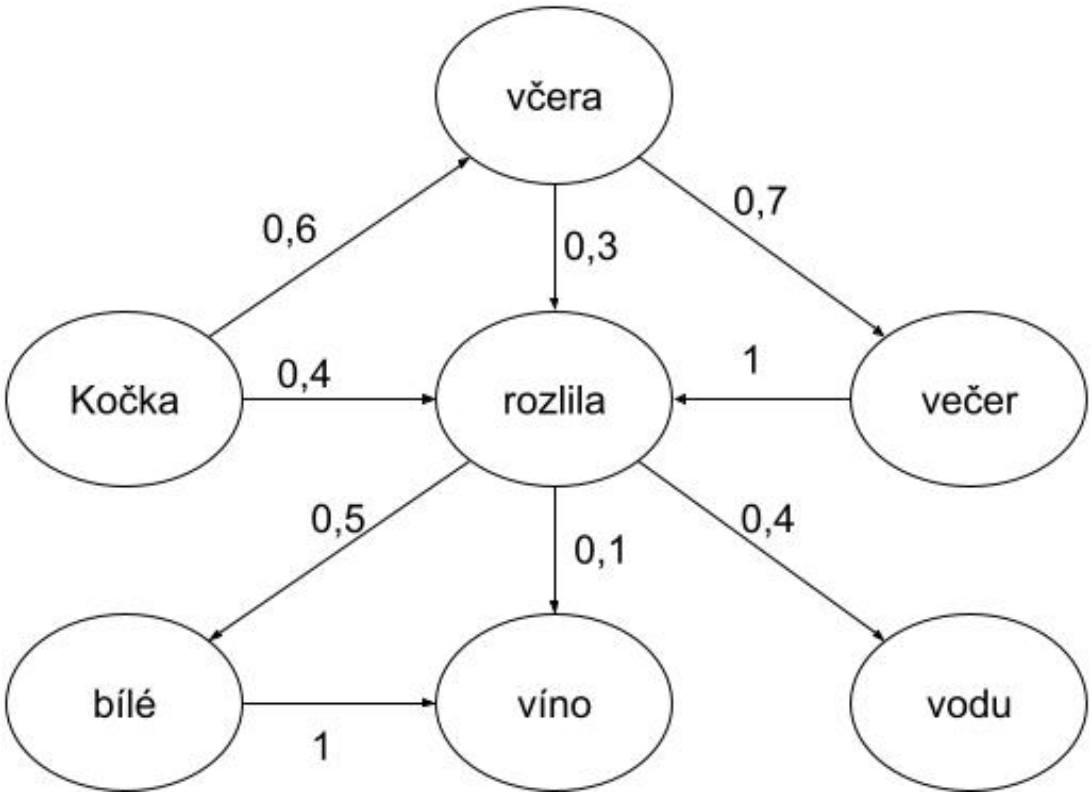
Tento princip vkládání znalostí do programu se vyskytuje i u programů založených na umělé inteligenci. Výzkumy však ukazují, že dlouhodobě lepších výsledků

umělá inteligence dosahuje bez pokusů vkládání lidských znalostí do algoritmů umělích inteligencí. Zde je přeložený výňatek ze článku The Bitter Truth od autora Rich Sutton.[15]

„1) Vědci z oboru umělé inteligence mnohokrát zkoušeli zabudovat znalosti do svých strojů,
2) toto vždy krátkodobě pomůže, a je uspokojující pro ony vědce, ale
3) v dlouhodobém měřítku to dosáhne na své maximum, ba dokonce zabrání budoucím pokrokům, a
4) průlomové posuny eventuelně přijdou opačným postupem, založeným na větší výpočetní síle, hledáním a učením. “

2.3.1 Markovův řetězec

Markovovy řetězce jsou jednoduché procesy, podle kterých se dá vygenerovat náhodný text. Tento stroj se skládá z množiny stavů a hran mezi jednotlivými stavy, pod jakou pravděpodobností může ze stavu jednoho, přejít do stavu druhého. Pokud si zvolíme, že jednotlivé stavy budou představovat slova, případně slovní spojení, získáme generátor textu. [4, 16]



Obrázek 3: ukázka Markovova řetězce

Tento řetězec může vygenerovat například „Kočka včera rozlila bílé víno.“

Je zde taky dobré zmínit, že Markovovy řetězce, i mnoho následujících algoritmů jsou lépe uzpůsobené pro generování anglického textu, než textu českého, protože v češtině se musí řešit věci jako skloňování a časování slov.

2.3.2 Bezkontextové gramatiky

Bezkontextové gramatiky jsou jeden z druhů formální informatiky. [4, 17] Bezkontextové gramatiky se skládají ze sady přepisovacích pravidel ve tvaru:

$$A \rightarrow \beta \tag{1}$$

„A“ v tomto případě představuje neterminál a beta řetězec neterminálů nebo terminálů. Zde je pro demonstraci příklad generující věty stejné jako v obrázku 3, přičemž procentní náhodnost nebude stejná.

$$\begin{aligned} S &\rightarrow Kočka\ R\ V \\ R &\rightarrow W\ rozlila \\ W &\rightarrow včera \mid včera\ večer \mid \epsilon \\ V &\rightarrow vodu \mid víno \mid bílé\ víno \end{aligned}$$

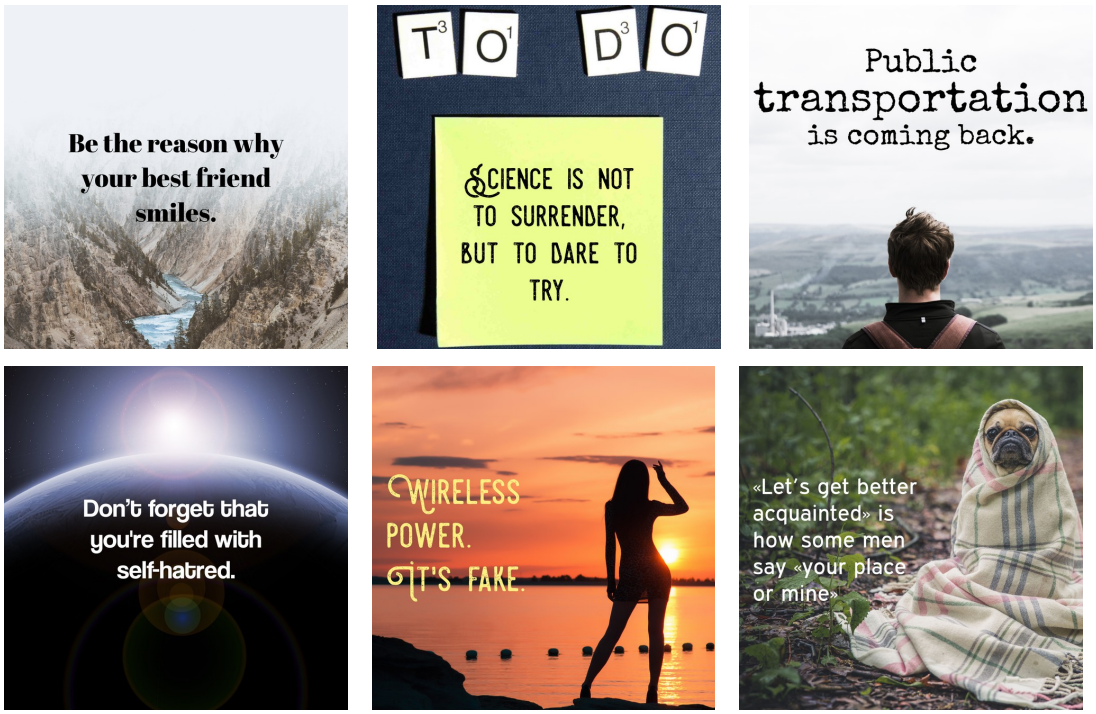
Symbol „|“ v bezkontextové gramatice představuje „nebo“, čili rozděluje možnosti, na které se může neterminál transformovat. Symbol epsilon zase představuje prázdný řetězec.

Narozdíl od Markovových řetězců zde lze zajistit, aby například osoba ve dvou po sobě jdoucích větách měla stejné jméno. Pro trochu komplexnější modely však narazíme na problém s různorodostí textu, smysluplností textu nebo několikanásobnou duplicitou v kódu.

Praktická část této práce se svým navržením podobá nejvíce tomuto konceptu, ale obsahuje více nástrojů, pro jednodušší generování.

2.3.3 Inspirobot

Inspirobot je webová stránka generující „motivační citáty“. Bohužel neuvádí přímo na jaké bázi generátor funguje. Píše jenom o tom, že je „umělá inteligence zaměřená na generování neomezeného množství speciálních inspiračních citátů pro obohacení bezvýznamné lidské existence“. [18]Ve skutečnosti však spíš jedná o algoritmus přímo napsaným člevěkem, než o umělou inteligenci. Při vygenerování dostatečného množství citátů si můžete všimnout určitých vzorců vět. Zde je ukázka několika citátů, co mě Inspirobot vygeneroval.



Obrázek 4: ukázky z generátoru Inspirobot

Tato stránka je zjevně navržena, aby generovala co nejhumornější obsah. Podoba s klasickými motivačními citáty je až druhotná. Ke každému citátu pak vezme jeden z přichystaných obrázkových šablon a font pro text. Inspirobot si

kolem sebe vytvořil svou komunitu[19] a vydělává prodejem triček, hrnků, plakátů a nálepek s těmito citáty[18].

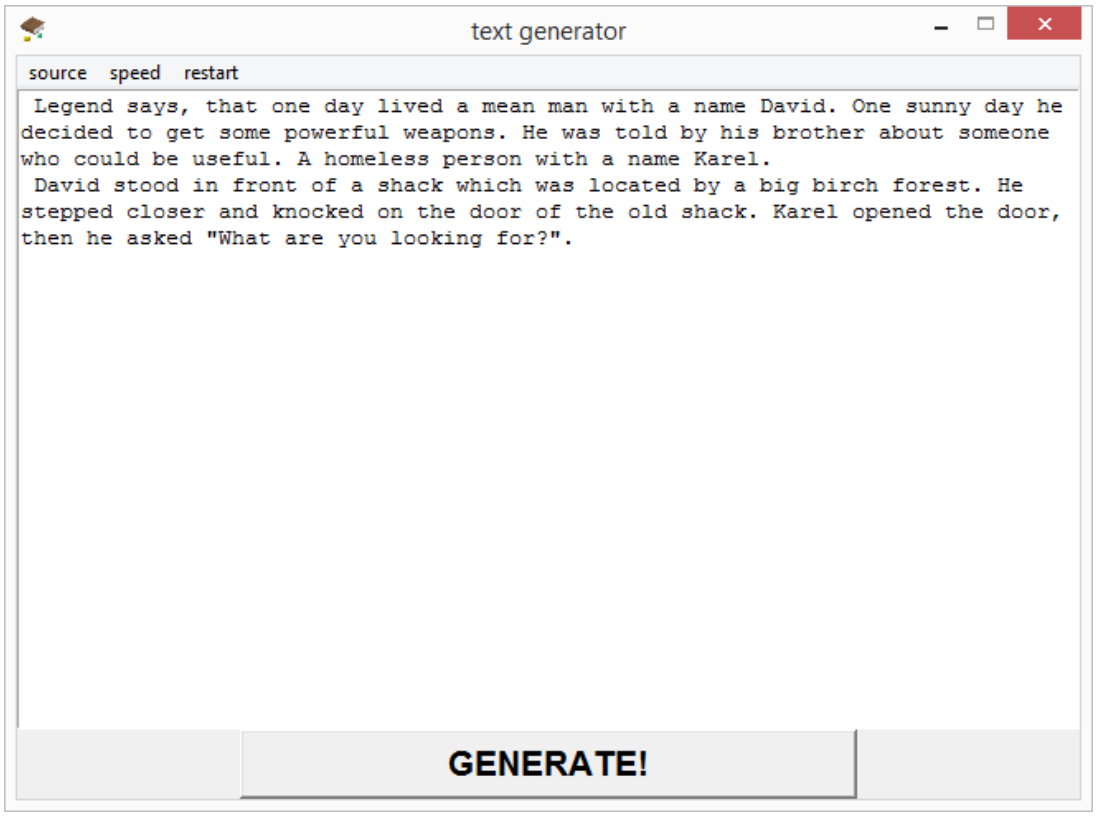
2.4 Zneužití generátorů textu

Generátory textu jsou v posledních letech čím dál více dostupné a tak je vhodné i mluvit o možnostech zneužití této technologie. [20] Vytvořením velkého množství lživých článků je nebezpečný způsob, jak ovlivnit myšlení občanů a kombinace s generátory falešných fotek a videí, může těmto lžím dodat iluzi pravdy. Samozřejmě, že všeho toho lze dosáhnout i lidskou silou, ale pro kvantitativně většího množství lživých článků je ekonomičtější zvýšit vypočetní sílu AI než zvýšit počet lidí pracujících na této dezinformaci.

3 Moje zpracování problematiky

Rozhodl jsem se vytvořit generátor na bázi pevně daného algoritmu s externími soubory pro data o generování. Tento přístup umožnňuje jednoduché rozšíření a úpravy. Algoritmus je stavěn na generování anglického textu. Český text má mnohem více pravidel kvůli skloňování a časování.

3.1 Vzled

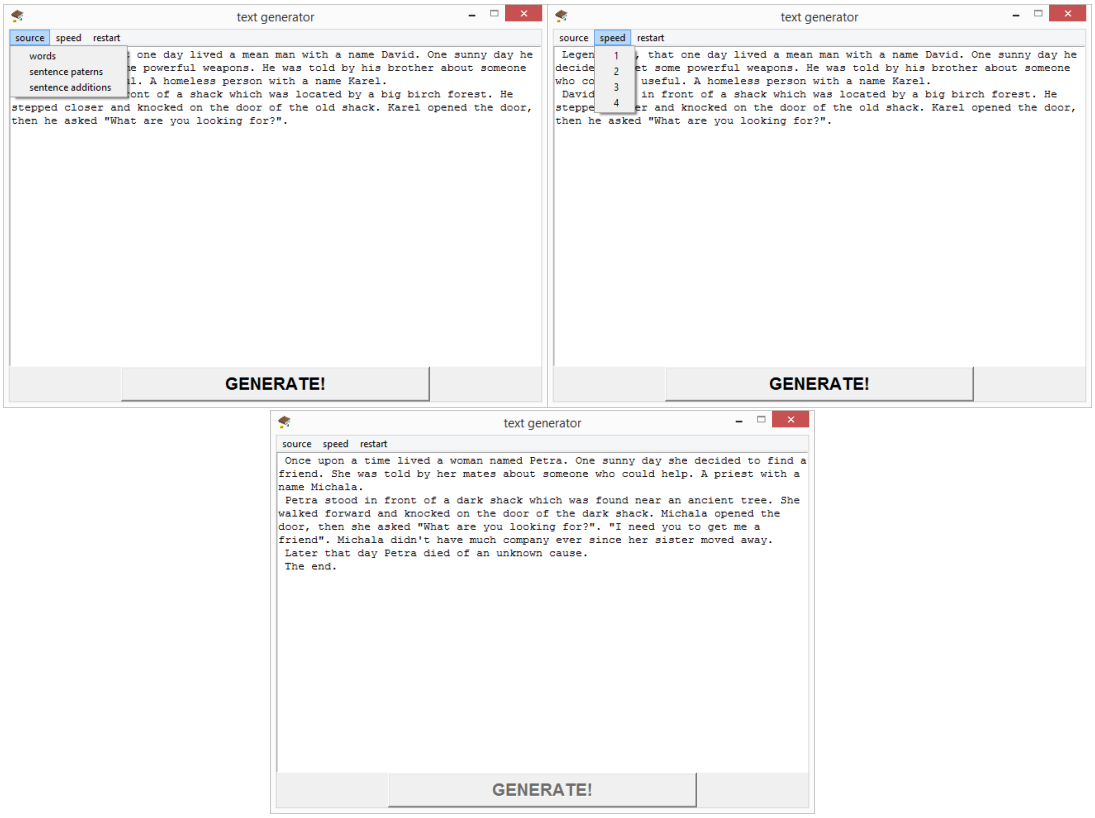


Obrázek 5: základní vzhled programu

Program je napsán v jazyce Python. Pro vyzualizaci jsem využil knihovnu Tkinter. Jedná se o jednoduché okno s tlačítkem pro generování textu.

V záložce „source“ se dá zvolit jiné soubory, které slouží jako zdroj pro generování textu. Záložka speed určuje rychlost generování textu. Program má čtyři rychlosti. Buď generuje při každém stisknutí tlačítka „GENERATE!“ znak, slovo, větu nebo 50 vět (případně míň, pokud už dopsal konec příběhu). Důvod existence této záložky je spíše pro mé osobní pobavení, než nějaký praktický účel. Jde akorát o to, abych mohl říct něco ve stylu „Co je tak těžkýho na tom, být spisovatel? Akorát mačkám jedno tlačítko a ono se to píše samo.“ Praktické využití jinak nemá.

Tlačítko „restart“ pak umožňuje generovat od začátku nový text. Pokud program dopíše příběh, tlačítko „GENERATE!“ se zdeaktivuje, dokud se neklikne na tlačítko „restart“.



Obrázek 6: záložky a zablokování tlačítka

3.2 Zápis dat

Pro můj generátor textu byla potřeba vytvořit způsob, jak do něj relativně efektivně vkládat nový obsah, který by program mohl generovat. Nedokážu si představit přehledný a efektivní způsob, jak tento mechanizmus zakomponovat přímo do kódu a vytvořit sadu tříd, seznamů nebo jiných datových struktur. Navíc je dobré dodržet pravidlo o segregaci logiky od dat. A tak jsem vytvořil vlastní primitivní jazyk, pro zápis obsahu pro generování textu.

Tento obsah jsem rozdělil do tří souborů. První soubor obsahuje slova, sousloví a případně krátké kusy vět, které jsou již dále nedělitelné a nemutovatelné. Druhý soubor obsahuje modely pro generování vět, jak na sebe mají navazovat, a co si z každé věty program do budoucna musí zapamatovat. Třetí soubor pak obsahuje části vět. Tento soubor je vhodný třeba pro věty vedlejší a odstranění duplicitní segmenty ve větách.

3.2.1 Soubor slov a slovních spojení

V tomto souboru je na každém řádku zápis jedné verzi slova, slovní spojení nebo nedělitelné části věty(dále jenom sousloví). Na každém řádku má před symbolem „/“ sousloví své identifikační jméno, kterým na něj referují zbylé dva soubory. Napravo od tohoto symbolu se pak nachází samotné sousloví.

```
1  in-front-of/in front of
2  in-front-of/before
3  house-adjective/old=an
4  house-adjective/wooden
5  house-adjective/dark
6  house/house
7  house/cottage
8  house/shack
9  that-had/that had
10 that-had/which had
11 that-had/which disposed with
```

Zdrojový kód 1: ukázka ze souboru slov a slovních spojení

Na řádku 3 si pak můžete všimnout ještě „=an“. [21]Toto je z toho důvodu, že v angličtině je neurčitý člen „a“ nebo „an“ je psán v závislosti na výslovnost slova, a nikoliv na jeho zápisu. Jelikož tento soubor neobsahuje výslovnost, je potřeba pro generování neurčitých členů mít uloženou aspoň informaci o správném neurčitém členu.

3.2.2 Soubor modelů vět

Na začátku je dobré uvést, že zápis do tohoto souboru je velmi citlivý na mezery.

```
1  helper-treasure/ " I plan a journey to journey-destination "
       answered name-man=person-help-name.helper-treasure2
```

Zdrojový kód 2: jednoduchý model věty ze souboru modelů vět

Jedna z možných vět, které tento model může vygenerovat, je například „"I plan an adventure to a cavern"said Mirek.“

V tomto souboru každý řádek odpovídá modelu jedné věty. Na začátku má věta své identifikační jméno oddělené lomítkem od zbytku věty. Mezi lomítkem a tečkou se nachází jména všech sousloví ve větě. Pro každé jméno algoritmus náhodně vybere jednu verzi onoho sousloví. Pokud se v souboru sousloví nenachází jediné sousloví s tímto názvem, algoritmus prostě vypíše onen název. Za tečkou se pak nachází možnosti všech následujících vět. (V ukázce je možnost pouze jedna.)

V této ukázce si můžeme všimnout, že uvozovky jsou brány trochu jako jméno sousloví, ale generátor dává mezery jenom z vnější strany uvozovek.

Dále si v ukázce můžeme všimnout „name-man=person-help-name“. „name-man“ Představuje jméno sousloví a „person-help-name“ je název proměnné v paměti. „name-man=person-help-name“ tedy znaméná, že algoritmus se podívá do své paměti, pokud se tam nachází proměnná „person-help-name“ tak algorit-

mus zapíše obsah té proměnné, a pokud ne, tak vygeneruje sousloví podle jména “name-man” a taky ho uloží do proměnné „person-help-name“.

```
1 helper-friend/name-man=person-help-name didnt have much human-
    interactions 50%ever since reason-for-no-friends=+.. later-
    conclusion
```

Zdrojový kód 3: model věty s procenty, koncem odstavce a odkazem na soubor s kusy vět

Jedna z možných vět které tento model může vygenerovat, je “Roman didn’t have a lot of human interactions ever since his brother moved away.”.

“50

“=+” znamená, že se nejedná o jméno sousloví, ale o jméno kusu věty.

Dvě tečky za větou místo jedné značí, že se jedná o konec odstavce.

```
1 helper-bad/name-man=name not-liked name-man=person-help-name because
     he=person-helper-pronoun reason-hate.(Jakub=person-help-name)
    helper-simpathetic helper-bad-ending
```

Zdrojový kód 4: model věty s podmíněným větvením

V této ukázce po větě mohou následovat dva modely vět a to „helper-simpathetic“ a „helper-bad-ending“. Ale „helper-simpathetic“ nastane pouze pokud v proměnné „person-help-name“ je uloženo jméno „Jakub“. Jde uvést i více podmínek za sebou ve formátu „(podmínka1)(podmínka2)...“.

3.2.3 Soubor větných částí

Zápis v tomto souboru je podobný tomu předchozímu. Chybí zde tečka ukončující větu a názvy následujících vět, protože pokračování je uvedeno v modelu věty, který si tuto větu vybral. Je zde naopak podpora pro více segmentů věty mající stejný název, pro umožnění větší rozmanitosti v rámci jedné věty.

```
1 reason-for-no-friends/his=person-helper-pronoun-owner someones died
    =+
2 reason-for-no-friends/his=person-helper-pronoun-owner someones moved
    -away 60%reason-move-away
3 reason-for-no-friends/he=person-helper-pronoun moved here some-time-
    ago
```

Zdrojový kód 5: ukázka modelů větných částí

3.3 Reprezentace souborů dat v programu

Data načtená z těchto třech souborů jsou přečtena a zpracována do několika úrovňových seznamů. Při spuštění programu jsou soubory sami zpracovány, ale program je může načíst znovu v případě zvolení v záložce “source”.

3.3.1 reprezentace souboru sousloví

```
1 [
2 ['start-time', ['once upon a time', 'legend says, that one day', '
     long time ago']],
3 ['start-place', ['in a mountain village', 'in the middle of a desert
     ', 'on a distant island']],
4 ...]
```

Zdrojový kód 6: část reprezentace souboru sousloví

```
1 start-time/once upon a time
2 start-time/legend says, that one day
3 start-time/long time ago
4 start-place/in a mountain village
5 start-place/in the middle of a desert
6 start-place/on a distant island
```

Zdrojový kód 7: odpovídající část v souboru

Ve vnější vrstvě seznamů každý prvek představuje jedno jméno sousloví a veškeré jeho verze výpisu. V druhé vrstvě je jméno uloženo na první pozici, a verze výpisu samotné jsou uloženy v seznamu na druhé pozici. Třetí vrstva pak tedy obsahuje ony jednotlivé verze.

Ve vedlejším souboru jsou pak uloženy všechny sousloví, před kterými se píše „an“ místo „a“.

3.3.2 Reprezentace souboru vět

První vrstva seznamu zde reprezentuje jednotlivé věty. V druhé vrstvě první položka reprezentuje jméno věty a poslední vrstva reprezentuje možné následující věty. Položky mezi nimi představují jednotlivá sousloví. Položka sousloví se skládá ze tří prvků. První je procentuální pravděpodobnost. Pokud v souboru nebyla uvedena, tak je jejich hodnota 100. Druhý prvek sousloví je jeho název a třetí je název proměnné, pokud nějakou přidruženou má. Ten samý prvek také může reprezentovat, že se jedná o jméno větné části namísto jméno sousloví. Je reprezentován znakem plus stejně jako v souboru.

```
[
['uvod-man',
[100, 'start-time', ''],
[60, 'start-place', ''],
[100, 'start-verb', ''],
[100, 'a', ''],
[60, 'person-adjective', 'person1-adjective'],
[100, 'man', 'person1-gender'], [100, 'named', ''],
[100, 'name-man', 'name'],
[0, 'he', 'person1-pronoun'],
[0, 'him', 'person1-pronoun-target'],
[0, 'his', 'person1-pronoun-owner'],
[30, 'name-rarity', '+'], ['(him=person1-pronoun-target)(his=person1
    -pronoun-owner)uvod-goal']
],...
]
```

Zdrojový kód 8: reprezentace jedné věty

```
uvod-man/start-time 60%start-place start-verb a 60%person-adjective=
    person1-adjective man=person1-gender named name-man=name 0%he=
    person1-pronoun 0%him=person1-pronoun-target 0%his=person1-
    pronoun-owner 30%name-rarity=+.(him=person1-pronoun-target)(his=
    person1-pronoun-owner)uvod-goal
```

Zdrojový kód 9: zápis této věty v souboru

Můžete si všimnout, že pár sousloví má procentuelní pravděpodobnost 0. To je proto, že je potřeba do paměti uložit jeho zájmena, ta ale se v samotné větě nevyskytují. (V souboru je ženská variace této věty, která vygeneruje ženské jméno a ženská zájmena).

Dále si na ukázce můžete všimnout, že podmínka následující věty je reprezentována ve stejném řetězci znaků, jako jméno oné věty. Podmínka je oddělena od jména v jiné části kódu. (Podmínka je zde uvedena z demonstrativních účelů a nemá logický význam, protože vždycky bude pravda.)

Při načítání tohoto souboru se taky ukládají do jiného seznamu jména všech vět, kterými může příběh začít. Ty jsou uloženy na prvním řádku souboru.

3.3.3 Reprezentace souboru částí vět

Zpracování částí vět se velmi podobá celým větám. Chybí zde přirozeně zmíňka o následující větě, protože následující text řeší ona věta, ve které se větná část nachází. Generátor navíc zde podporuje více větných částí se stejným jménem a mezi nimi vybírá náhodně.

```
[
['house-details', [100, 'that-had', ''], [100, 'many', ''], [100, '
    house-details', '']], ['house-details', [100, 'which-was', ''],
    [100, 'located', ''], [100, 'house-place', '+']],
...]
```

Zdrojový kód 10: reprezentace dvou verzí jedné části věty

```
house-details/that-had many house-details
house-details/which-was located house-place=+
```

Zdrojový kód 11: jejich zápis v souboru

3.4 Výpis

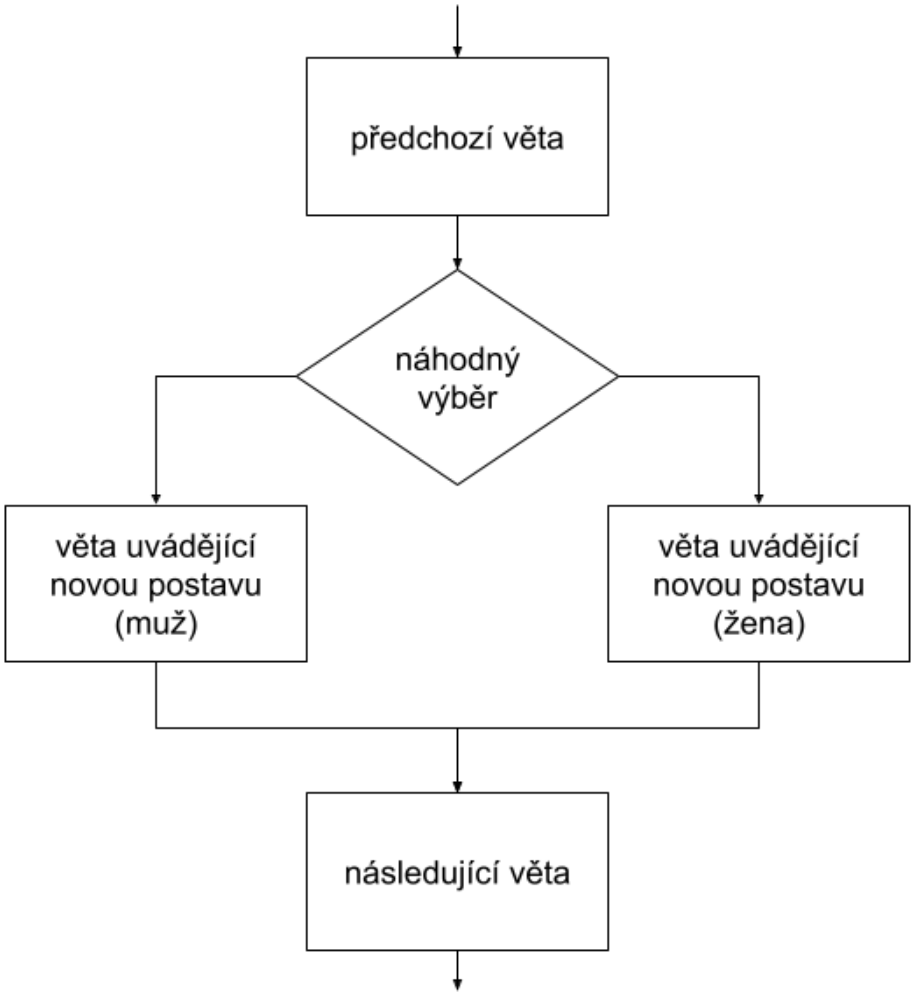
Podle zvolené rychlosti vypisování se po každém zmáčknutí tlačítka „GENERATE!“ vypisuje jedno písmeno, jedno slovo, jedna věta nebo až 50 vět (určeno pro vypsání celého textu). Interně je toto řízeno tak, že si program načte model následující věty, dosadí si za včechny názvy sousloví a seskládá tak větu. Následně tu větu upraví tak, aby na začátku bylo velké písmeno, na konci tečka, upraví slova „a“ na „an“, které jsou potřeba, upraví správně mezery a, pokud to model uvádí, přidá zalomení pro konec odstavce. Tato věta je pak uložena v zásobníku, ze kterého se text postupně vypisuje do textového pole. Při rychlosti jedna vybírá písmeno po písmenu. Při 2 vypisuje, dokud nenarazí na mezeru nebo konec. Při 3 vypíše rovnou celý zásobník. Rychlost 4 pak opakovaně vypisuje a naplňuje zásobník. V případě, že model věty nemá žádný navazující model, dojde k zablokování tlačítka „GENERATE!“

3.5 Zhodnocení

Co se týče potenciálních rozšíření pro tento algoritmus, tak na prvním místě zmíním editování vstupních souborů. Jedna z velkých výhod vývojových prostředí pro vývojáře je ten, že ono prostředí barevně rozlišuje jednotlivé části kódu, a tak ten kód zpřehledňuje. Jelikož vstupní soubory „paterns.txt“, „additions.txt“ a „words.txt“ používají svoji vlastní syntaxi, žádné prostředí nezbarvuje tento kód, což znepřehledňuje práci. Z mojí zkušenosti je asi nejvhodnější PSPad editor, protože ten po označení slova zvýrazní všechny výskyty bez nutnosti využití klávesové zkratky pro hledání výskytů slova.

Další věc, která by zrychlila přidávání obsahu, je zakódovat do programu gramatiku pohlaví. Řekněme, že chceme větu, která do příběhu přidá novou postavu. Chceme, aby tahle věta někdy přidávala mužskou a někdy ženskou postavu. Tato postava se bude vyskytovat i v jiných větách, takže jejich pohlaví musí být taktéž někde uloženo. V obrázku 7 je znázorněné současné řešení tohoto problému. Tyto dvě věty si pak vkládají do paměti všechna zájmena, kterými na ni má referovat

(he, him, his nebo she a v budoucnu by se dali přidat i himself a herself). Tímto způsobem ovšem vzniká redundantní kód, jelikož u této věty existují dvě varianty (a měli bysme tři, pokud bysme chtěli příběhy rozšířit i o postavy co, používají zájmena they, them, their), a navíc se každé zájmeno ukládá zvlášť.



Obrázek 7: Současná implementace pohlaví

Podle mě nejvhodnější řešení by bylo ukládat postavy do své vlastní datové struktury. Přiřazování zájmen a také jmen by pak mohlo probíhat automaticky bez potřeby rozepisování všech zájmen a jmen ve vsupních souborech. Nabízí se zde pak i možnost ukládání i dalšího volitelného obsahu, který by se jinak ukládal do seznamu ostatních údajů. Například že je postava barvoslepá by se ukládalo a přistupovalo podobně, jako ke jménu. Šlo by zde tedy o princip tříd a jejich vlastostí, jak se používají v mnohých programovacích jazycích.

3.6 Využití

Můj generátor popsaný v této práci vytváří příběhy podle předdefinovaných modelů, slov a ostatních požadavků popsaných v anotaci na úvodu této práce, avšak jeho různorodost není nijak úchvatná. Technicky jednoduchým, ale časově velmi náročným řešením tohoto problému je zvýšit množství větných modelů a s tím i rozšířit zbylá vstupní data. Větší různorodosti s objemově stejným množstvím vstupních dat lze také dosáhnout soustředěním se na kratší texty a kladením menšího důrazu na fakticky korektní část textu.

Praktické využití mé technologie by dle mého pohledu tedy nemělo být vygenerování několika stránkových příběhů, ale je vhodný pro vypisování krátkých textů, jako například webová stránka Inspirobot, nebo v herním průmyslu pro generování konverzací a textů uvnitř nějaké hry s náhodně generovaným obsahem.

Závěr

Pro vytvoření komplexního generátoru příběhů se skutečně náhodným obsahem je vhodné využít technologie strojového učení. Výzkum v tomto odvětví dosáhl již takového pokroku, že překonal člověkem vytvořené algoritmy ve své různorodosti a není přitom limitován svým chápáním gramatiky a našeho světa. Generátory textu bez strojového učení však nadále mají své místo. Jejich výhody jsou jednoduchost vývoje a větší kontrola nad generovaným obsahem.

Praktická část této práce obsahuje mé zpracování této problematiky pomocí algoritmů, bez využití strojového učení. Algoritmus generuje podle tří textových souborů které obsahují informace pro generování zapsané ve své vlastní syntaxi. Algoritmus je napsán tak, aby tyto vstupní soubory mohly být upraveny a rozšířeny o další obsah, pro obohacení různorodosti příběhů.

Conclusions

For development of complex story generators with truly random content it is strongly beneficial to use machine learning technology. The research in this field has reached the point where it can made better texts than any human made algorithms without machine learning, while also not being limited by its understanding grammar and our world. Text generators without machine learning still have their benefits, such as simplicity of development and better control over the output.

My generator described in this document produces stories according to its input sentence patterns, words and other things described in the annotation, but with limited creativity. Technically simple but very timely solution would me to increase the volume of input data. Bigger diversity of results with same input volume can also be achieved by focusing on generating shorter texts and less focus on whether the text will be factually correct.

Practical use of my technology therefore should most likely not be generating several pages long story, but it is useful for writing short texts, like the website Inspirobot, or in gaming industry for creating dialogs or texts in a game with randomly generated content.

A Obsah přiloženého CD

Přiložené CD obsahuje zdrojový kód programu a zkompilovanou verzi pomocí pyinstaller do .exe souboru.

`text_generator/generator.py`
: Zdrojový kód programu obsahující veškerou logiku.

`text_generator/paterns.txt`,

`text_generator/additions.txt`,

`text_generator/words.txt`
: Tyto tři soubory představují data pro generování textu. Pro upravení generování můžete editovat tyto soubory nebo vytvořit nové podle jejich vzoru a po spuštění programu nastavit cestu k vašim novým souborům v záložce „source“.

`text_generator/generator.exe`
: Spustitelný program v systému Windows. Je možné, že antivirus bude mít problém se spuštěním. Je potřeba povolit vyjímku.

`about.pdf`
: Kopie této práce.

U veškerých cizích převzatých materiálů obsažených na CD/DVD jejich zahrnutí dovolují podmínky pro jejich šíření nebo přiložený souhlas držitele copyrightu. Pro všechny použité (a citované) materiály, u kterých toto není splněno a nejsou tak obsaženy na CD/DVD, je uveden jejich zdroj (např. webová adresa) v bibliografii nebo textu práce.

Seznam zkratek

AI Artificial Inteligence

GRU Gated Recurrent Units

Lipsum Lorem ipsum

LSTM Long-Short-Term Memory

NLP Natural Language Processing

RNN Recurent Neural Network

Literatura

[1] Generativní umění. https://www.netinbag.com/cs/internet/what-is-generative-art.html.

[2] Neuroflash - jak AI generátory textu změní průmysl. https://www.datarobot.com/blog/what-is-natural-language-processing-introduction-to-nlp/.

[3] Lorem ipsum. https://cs.wikipedia.org/wiki/Lorem_ipsum.

[4] KADLEC, Bc Jan. Automatický generátor textu. [online]. [Cit. 2022-5-31]. https://dspace.tul.cz/bitstream/handle/15240/12113/mgr_23293.pdf.

[5] DataRobot - přirozené zpracování jazyka. https://neuroflash.com/blog/ai-text-generator/.

[6] Kapitola z AI generované knihy o Harry Potterovy. https://botnik.org/harry-potter-chapter/.

[7] OpenAI. https://openai.com/.

[8] DALL · E 2. https://openai.com/dall-e-2/.

[9] film Sunspring. https://youtu.be/LY7x2Ihqjmc.

[10] Distill - Vyzualizace paměti v RNN. https://distill.pub/2019/memorization-in-rnns/.

[11] Wikipedie - RNN. https://en.wikipedia.org/wiki/Recurrent_neural_network.

[12] Wikipedie - LSTM. https://en.wikipedia.org/wiki/Long_short-term_memory.

[13] Wikipedie - GRU. https://en.wikipedia.org/wiki/Gated_recurrent_unit.

[14] nested LSTM. https://arxiv.org/pdf/1801.10308.pdf.

[15] The Bitter Truth. http://www.incompleteideas.net/IncIdeas/BitterLesson.html.

[16] Markovovi řetězce. https://cs.wikipedia.org/wiki/Markov%C5%AFv_%C5%99et%C4%9Bzec.

[17] Bezkontextová gramatika. https://cs.wikipedia.org/wiki/Bezkontextov%C3%A1_gramatika.

[18] Inspirobot. https://inspirobot.me/.

[19] Reddit fórum pro komunitu Inspirobota. https://www.reddit.com/r/inspirobot/.

[20] risk zneužití generátorů textu. https://arxiv.org/pdf/1906.01946.pdf.

[21] Cambridge - pravidla psaní "a"a "an"v angličtině. https://dictionary.cambridge.org/grammar/british-grammar/a-an-and-thel.